

LA-UR-21-28423

Approved for public release; distribution is unlimited.

Title: Summer 2021 Internship Report

Author(s): Mahanama, Rajith Bahanuka

Intended for: Report

Issued: 2021-08-23

Disclaimer:

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy under contract 89233218CNA000001. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Summer 2021 Internship Report

Name	Bhanuka Mahanama
UIN	01160805
Email	bhanuka@cs.odu.edu
Course	CS669 Practicum
Date	August 17, 2021

Section I: Position Information

- **Title:** Summer Research Intern
- **Organization:** Los Alamos National Laboratory, Los Alamos, NM
- **Department:** Research and Prototyping Team, Research Library
- **Supervisor:** Dr. Lyudmila Balakireva

About the Organization

Los Alamos National Laboratory is a Federally-funded research and development center with the mission of solving national security through scientific research and development. The research areas of the laboratory include nuclear security, national defense, energy, counter-terrorism, and the environment. The laboratory hires approximately 2000 students each year to work on scientific research and development projects as a part of their summer internship program. During the internships, the students get to work on a project to gain experience in research and development.

The library's research and development arm pursue research on various aspects of information infrastructure in the digital age. The Library Research and Prototyping team (Proto Team) under the research library explores aspects of scholarly communication, covering aspects of infrastructure, interoperability, and persistence. One of the key contributions of the team is the contribution in standardizing and forming the infrastructure for Mementos in web archiving. The Memento framework in web archiving enables access to digital resources in prior versions serving both persistence and the interoperability of the information.

About the Project

During the internship program, I worked on improving the Memento infrastructure by updating the memento validator. The Memento validator provides the functionality of testing the compliance of the web resources to the Memento specification. The application is intended to be used by the community at large such as web archives, researchers, librarians, and general users, with different levels of technical expertise.

The Memento protocol serves as the framework for providing time-based access to resource states over Hyper-Text Transfer Protocol (HTTP). Using memento protocol, the user can negotiate the content in the time dimension, such as requesting the resource at a specific point in time. The memento protocol defines the four types of resources and standards for each type of resource to ensure interoperability. As a result, the compliance of resources to the standards plays a vital role in the smooth functioning of the Memento web.

Section II: Duties/Responsibilities

During the internship, I was responsible for carrying out the entire project, including requirement analysis, design, development, deployment, and testing. In addition, I was also responsible for creating both technical and non-technical documentation for the project to ensure knowledge transfer at the end of the internship.

As a part of the onboarding process, I had to attend a series of orientation programs in the first week of the internship. The orientation programs provide the opportunity to familiarize the working environment and the internal processes in the laboratory.

From the first week onwards, I had weekly one-to-one meetings with my supervisor and sometimes accompanied by my co-supervisor. The primary purpose of the weekly meetings was to report the progress, obtain feedback, resolve issues, and report plans. In addition, there were bi-weekly team meetings where the entire Proto team attended. During the team meetings, all team members presented what they worked on, their plans for the upcoming weeks, and the team-wide announcements. Since all the members of the team were teleworking, all the meetings were held through teleconferencing tools.

The project's first task was to familiarize me with the concepts in memento validation and the existing applications. Then I went through the source code of the current software and explored the possibility of improving the validators based on the existing code. Based on an analysis, I had presented ideas for enhancing the application and the plan for the project. This allowed me to apply my knowledge in programming, data structures, software engineering, web programming, and software development.

Upon the presenting improvements to the validator, I had to complete the development and testing of the application. There, I had to decisions on technology selection, modularity of the application, Application Programming Interface (API) development, and web application development. During these decisions, I had to address the community's requirements at large and possible extensions to the application in the future.

As the project's next steps, I started by developing a core library for testing Memento validation. Then by utilizing the core library, I created an HTTP API, Command Line Interface, and a daily validator script as series of applications. Finally, I developed a web interface that allows general users to access the Memento validation, consuming the HTTP API. At the end of development, I packaged the entire application and provided deployed into laboratory servers for general user access.

In addition to the opportunities to expand knowledge within the team and the technical area, LANL provides a wide access variety of workshops and sessions through the internal network. Different groups organizing these sessions help understand how other teams are working towards the laboratory's mission.

Section III: Progression

Before the internship program, I had a limited understanding of the Memento framework and the role of the validator as a part of the Memento infrastructure. Throughout the project, I gained a deeper understanding of the Memento framework and the Memento validator. The more profound knowledge of the memento framework was also helpful in forming the project plan to create sub-tasks within tasks in the project. Further, comments and ideas from the advisors helped me set the goals and deliverables at the end of the project.

With the continuation of the project, my advisor delegated me additional responsibilities such as creating a library for link-header parsing and expanding the testing scope of the application. Further, I assumed the responsibility of ensuring the portability of the application in different platforms and the thorough documentation for the application. Through this, I was able to

understand various technical and non-technical considerations during application development helpful in improving my graduate research.

Section IV: Academic Relevance

Coursework

The project's first task was to familiarize me with the Memento framework and the existing validator applications to identify any weaknesses in the current validator. Based on the initial analysis of the existing validator source, I found that the validator has limited modularity and provides validator functionalities only through the web interface. Further, the application uses Python 2, discontinued by the Python software foundation and the community.

In contrast, the daily validator application had a different code base independent of the web validator, even though both served a similar functionality. It used a dedicated configuration file aside from the aggregator configuration despite sharing the same archives list.

Based on these findings, I decided to restructure validator applications for a shared codebase and resources. As the solution, I decided to develop a core library for performing Memento validation. The core library provides functionalities in levels of granularity. Firstly, it includes validation at the resource level, such as validating a URI-R for compliance with URI-R specifications. Secondly, it provides attribute level validation, such as validating that URI-R has a valid TimeGate in the link header. Then each validator application will consume the core library and perform accordingly. In this manner, the codebase will be shared and modular, with easier maintenance.

In the case of shared configurations between the daily validator and the aggregator, we added additional fields to the aggregator configuration to be used by the validator.

When implementing the plan above, I ran into two challenges. Firstly, the classification of the tests is based on some common attributes between tests. For this, I considered the type of HTTP header being considered and the attribute within the header. For instance, I added all the tests relating to the link-header memento relationship to a single module. Secondly, there was no standardized way to parse the link headers in an HTTP request. For this, I went through the existing application and a few other pre-existing related applications (MemGator and Aggregator) and implemented the link parsing mechanisms used.

Professional Literature

The Memento protocol[1] is defined in the Internet Engineering Task Force Request for Comments (IETF RFC) 7089 as a content negotiation mechanism in the time dimension. The Memento protocol operates using HTTP and provides DateTime negotiation through linked resources. The Memento protocol defines four types of resources as 1. Original (URI-R), 2. memento (URI-M), 3. TimeGate (URI-G), and 4. TimeMap (URI-T).

The original web resource is a web resource for which a user needs to find a past version. For instance, www.example.com, as it appears at present, www.example.com can be considered an Original resource. The Original resource is subject to continuous changes as the web resource evolves. A snapshot of the Original resource taken at a specific point in time is a Memento. As a result, a Memento represents how the Original resource looked like at some point in the past. For a given Original resource, there can be many Mementos reflecting different states of the Original Resource.

The TimeGate is a resource that provides access to the previous states of the Original resource through DateTime negotiation, meaning TimeGate decides which resource most resembles the

Original resource for the given DateTime. For communicating the desired DateTime to the TimeGate, the client uses the Accept-Datetime HTTP header.

A TimeMap resource provides links to all the stored states for a resource along with the corresponding timestamp. As a result, the TimeMap resources help find all the prior states of a resource instead of traversing one by one. A TimeMap resource or a Memento resource can inform the user about the available TimeMap resource using the Link header with the relationship type TimeMap.

Even though the Memento Protocol enables the traversal in the time domain easily, fewer tools focus on this aspect. A tool aggregates mementos in general, known as an Aggregator, and based on the implementation of the Aggregator, the performance and the features can vary. For instance, depending on the archives the Aggregator uses for the aggregation process, the results return by aggregators can change.

The MemGator[2] is an open-source, portable, concurrent aggregator written in the Go language. Since the aggregation process involves communication with multiple archives, non-concurrent processing can significantly impact the performance. The MemGator overcomes the problem using multiple processes to speed up the operation. For the implementation of the idea, MemGator uses lightweight processes in Go called "goroutines." Each process will fetch and parse the TimeMap and sends the results to a collecting process. The collecting process aggregates the results from the processes. For implementing the communication between the collecting process and the TimeMap parsing processes, MemGator uses Go channels.

In the evaluation of the MemGator, the authors test the MemGator for different stress levels. As the experiment results, the authors claim to observe better throughput with increasing stress levels due to the proposed methodology's efficient resource utilization.

Section V: Future Projections

Before the internship, I had no experience working in the United States outside the on-campus university employment. I experienced the working environment, processes, and culture in a United States national laboratory through the internship. The experience unlocks me with many career opportunities in research and development, such as research and development engineer, data scientist, system engineer, and research scientist. One of my observations was that a comprehensive understanding of a standard or a framework is vital in obtaining a career in a field like web archiving.

I updated my homepage, LinkedIn, resume, and curriculum vitae by including experience from the internship. Further, during the project, I also identified some improvements for the project. Additionally, I got the opportunity to expand the network by collaborating with researchers at the laboratory.

Section VI: Conclusion

This was my first internship in the United States as a Ph.D. student, and I'm grateful for the opportunity. Even though the program was remote, the team was supportive, open to suggestions, and understandable. I firmly believe the experience I gained by working with a team of scientists helped me think and critically evaluate decisions and help complete my graduate program.

I am grateful to my internship supervisor Dr. Balakireva and Dr. Klein, for recommending and selecting the internship program.

Section VII: Beneficial Suggestions

Publishing the mementoweb validator package into pip and making the project opensource. Currently, all the applications and the core code reside in the same directory, a suggestion is to move core code to separate repository and have separate repositories for each application.

Unifying the documentation: Currently documentations exist in both google docs, auto generated code documents, and GitHub documentations. A suggestion is to merge all into one format.

Update aggregator and the proxies and containerize them.

Section VIII: Quotes / Photos / Videos

<Image from mini symposium>

Figure 1: Teleworking for LANL

"I'm grateful for the opportunity I got to work as a summer research intern at LANL. Even though the team was teleworking, the team was supportive, open to suggestions, and understandable. I

strongly believe the experience I gained through the internship will help me in my future in completing my graduate program and in my future career decisions.”

References

- [1] Van de Sompel, Herbert, Michael L. Nelson, and Robert Sanderson. "HTTP framework for time-based access to resource states—Memento." Rfc7089, IETF (2013).
- [2] Alam, Sawood, and Michael L. Nelson. "MemGator—A portable concurrent memento aggregator: Cross-platform CLI and server binaries in Go." 2016 IEEE/ACM Joint Conference on Digital Libraries (JCDL). IEEE, 2016.